

## Durham Research Online

---

### Deposited in DRO:

08 May 2017

### Version of attached file:

Published Version

### Peer-review status of attached file:

Peer-reviewed

### Citation for published item:

Lin, Bao Lin and Sun, Xiaoyan and Salous, Sana (2016) 'Solving travelling salesman problem with an improved hybrid genetic algorithm.', *Journal of computer and communications*, 4 (15). pp. 98-106.

### Further information on publisher's website:

<https://doi.org/10.4236/jcc.2016.415009>

### Publisher's copyright statement:

Copyright © 2016 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).<http://creativecommons.org/licenses/by/4.0/>

### Additional information:

---

## Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in DRO
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full DRO policy](#) for further details.

# Solving Travelling Salesman Problem with an Improved Hybrid Genetic Algorithm

Bao Lin<sup>1,2</sup>, Xiaoyan Sun<sup>1</sup>, Sana Salous<sup>3</sup>

<sup>1</sup>School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, China

<sup>2</sup>School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, China

<sup>3</sup>School of Engineering and Computing Sciences, Durham University, Durham, UK

Email: baolin\_zj@163.com, xysun78@126.com, sana.salous@durham.ac.uk

**How to cite this paper:** Lin, B., Sun, X.Y. and Salous, S. (2016) Solving Travelling Salesman Problem with an Improved Hybrid Genetic Algorithm. *Journal of Computer and Communications*, 4, 98-106.  
<http://dx.doi.org/10.4236/jcc.2016.415009>

**Received:** October 25, 2016

**Accepted:** November 25, 2016

**Published:** November 28, 2016

## Abstract

We present an improved hybrid genetic algorithm to solve the two-dimensional Euclidean traveling salesman problem (TSP), in which the crossover operator is enhanced with a local search. The proposed algorithm is expected to obtain higher quality solutions within a reasonable computational time for TSP by perfectly integrating GA and the local search. The elitist choice strategy, the local search crossover operator and the double-bridge random mutation are highlighted, to enhance the convergence and the possibility of escaping from the local optima. The experimental results illustrate that the novel hybrid genetic algorithm outperforms other genetic algorithms by providing higher accuracy and satisfactory efficiency in real optimization processing.

## Keywords

Genetic Algorithm, Hybrid Local Search, TSP

## 1. Introduction

Travelling Salesman problem (TSP) [1] is a well-known nondeterministic polynomial time (NP-Hard) problem and a typical example in combinatorial optimization researched in both operations research and theoretical computer science. TSP can be described as: Given a finite set of  $n$  cities and a cost matrix  $D = [d_{ij}]$ , in which  $d_{ij}$  denotes the cost (e.g., distance, money, time) of traveling from city  $i$  to city  $j$  ( $i, j \in \{1, 2, \dots, n\}$ ), TSP aims to find the optimal tour  $\pi = (\pi_1, \pi_2, \dots, \pi_n)$  that visits each city exactly once and eventually return to the starting city. The objective function of this linear programming problem in TSP is minimizing the sum value of the cost in

the valid tour:  $\min \sum_{i=1}^{n-1} d_{\pi_i, \pi_{i+1}} + d_{\pi_n, \pi_1}$ .

TSP and its variants transpire in a large variety of real-world applications such as vehicle routing, scheduling problems, integrated circuits designs, graph theory, gene ordering, and so on. Up to now, many literatures have reported that lots of techniques applied to those problems suffer from exponentially growing complexity for finding out an optimal solution. Various heuristic approaches have been conducted to obtain the global optimum or reasonable suboptimal solutions in solving TSP, such as Genetic Algorithm (GA) [2] [3], Simulated Annealing (SA) [4], Ant Colony Optimization (ACO) [5], Particle Swarm Optimization (PSO) [6], Tabu Search (TS) [7], Neural Network (NN) [8], and so on. Among all the evolutionary algorithms, GA has paid more attention for its effectiveness compared to other algorithms in solving discrete combinatorial optimization problems. Furthermore, there are many researches [9] [10] [11] [12] on combining of GA with other evolutionary algorithms to improve accuracy and efficiency in solving TSP.

In this paper, a novel hybrid GA (HGA), which incorporates the local search method into GA, is proposed to be as an effective heuristic approach for the two-dimensional Euclidean TSP. The proposed algorithm has the trade-off of preserving population diversity and preventing the premature convergence in the evolutionary processing. Some mechanisms are specifically designed and constructed to improve the quality of the final solutions with reasonable time-consuming for TSP, such as the elitist choice strategy, the local search crossover operator and the double-bridge random mutation. The proposed HGA is verified with some standard data benchmarks from TSPLIB [13]. In this work, we make the following contributions: 1) We introduce hybrid version of both algorithms with GA and the local search method as a promising method for TSP; 2) We conduct the experimental research to validate the effectiveness and efficiency of the HGA by a comprehensive comparison over other evolutionary algorithms.

The rest of this paper is organized as follows. Section 2 briefly introduces the traditional GA for TSP. Section 3 presents the description of the proposed HGA. Section 4 demonstrates the experimental results. Finally, Section 5 summarizes the paper and derives conclusions with the scope of future extension of this work.

## 2. Background

GA is an optimization model inspired by Darwinian evolution and was first proposed by John Holland in 1975 [2]. The traditional GA is an adaptive search technique to find the global optimum or quasi-optimum for optimization problems. The main operations using GA to solve TSP can be described as follows. Generally GA starts from a population of encoded feasible solutions, followed by evaluating the fitness function of individuals in the population. GA produces the next generation from the previous generation with three operations to exploit the search space, namely selection, crossover, and mutation operators. Selection operation typically chooses competitive individuals from the mating pool according to the survival strategy, such as roulette wheel selection, tournament selection, etc. Crossover and mutation operations primarily provide good searching diversification in the solution space.

The basic steps of GA are presented:

Procedure of GA
1 Population Initialization
2 Repeat
3 Parents selection process
4 Crossover operator
5 Mutation operator
6 Survivor selection process
7 Until termination criteria reached
8 Output the final results
End

In solving the small-scale TSP, the traditional GA can provide the satisfactory optimal solution with an acceptable time-consuming. However, when the number of city nodes increases, the traditional GA is greatly challenged with respect to efficiency because of the stagnation behavior and premature convergence. Therefore, designing the effective operators in GA is considered to be a promising way to solve the medium-scale or large-scale TSP.

### 3. The Proposed Algorithm

In this paper, a novel heuristic HGA, which is a mechanism of combining GA with the local search strategy, is proposed to efficiently solve TSP. The proposed HGA incorporates the local search strategy into GA by the novel updating strategy which is the combination of the elitist choice strategy, the local search crossover operator and the double-bridge random mutation. It takes full advantage of exploration ability of GA and exploitation capability of the local search method to improve the quality of the optimum or sub-optimum solutions with reasonable time-consuming.

#### 3.1. Initialization

In permutation encoding of TSP, the individual chromosome representing a feasible solution is encoded as a  $N$  dimensional integer vector  $\pi_l = (\pi_{l1}, \pi_{l2}, \dots, \pi_{ln})$  (where  $\pi_{l1}, \pi_{l2}, \dots, \pi_{ln}$  represent  $N$  consecutive city nodes in a complete valid tour with  $l = 1, 2, \dots, M$  ( $M$  is the population size)). An initial population of high quality can accelerate the population evolutionary processing of HGA to reach a satisfactory global optimum or suboptimal solutions. The generating process of the initial population in HAG is as follows: The 80% individuals of the initial population are randomly generated by random number generator and the other 20% individuals are created by the heuristics method based on the greedy strategy. The generating method of the initial population not only maintains the population diversity but also speed up convergence on the final solution by providing promising initial solutions.

### 3.2. Selection Operator

The mechanism of combining roulette wheel selection and elitist choice strategy is used as selection operator in HGA. These survival chromosomes with the best fitness value will have higher probabilities to be selected from the matting pool of the previous generation. In this paper, the fitness function is used as Equation (1).

$$F(\pi) = \frac{1}{1 + f(\pi)} \quad (1)$$

### 3.3. Local Search Crossover Operator

Crossover operator is a vital factor which determines the characteristic optimization of HGA. Local search crossover operator, which preserves part of the good edge knowledge of the selected parent chromosomes obtained with the previous population, improves effectiveness and efficiency of HGA by exploring the more promising different regions of the solution space for TSP. The mechanism of the local search crossover operator is described in the following:

1) Randomly select two parent individuals (saying  $\pi_{pi} = (\pi_{pi1}, \pi_{pi2}, \dots, \pi_{pin})$  and  $\pi_{pj} = (\pi_{pj1}, \pi_{pj2}, \dots, \pi_{pjn})$ ) from the matting pool of the previous generation according to crossover probability  $P_c$ .

2) From the first parent individual  $\pi_{pi}$ , one gene node is selected randomly as the starting node of the offspring chromosome  $\pi_{qi}$ . For example, the node  $\pi_{pia}$  is selected as the start point of the new individual, then all the one-step neighborhood gene nodes of  $\pi_{pia}$  in both the two parents individuals are determined and they are  $\pi_{pi(a-1)}$ ,  $\pi_{pi(a+1)}$ ,  $\pi_{pj(b-1)}$ ,  $\pi_{pj(b+1)}$  (the value of the node  $\pi_{pjb}$  in  $\pi_{pj}$  is equal to that of the node  $\pi_{pia}$  in  $\pi_{pi}$ ). Comparing the travelling distances between the current node and those one-step neighborhood gene nodes ( $d_{\pi_{pia}, \pi_{pi(a-1)}}$ ,  $d_{\pi_{pia}, \pi_{pi(a+1)}}$ ,  $d_{\pi_{pia}, \pi_{pj(b-1)}}$ ,  $d_{\pi_{pia}, \pi_{pj(b+1)}}$ ) respectively, the nearest node to the current node in the neighborhood is selected and inserted into the sequence of  $\pi_{qi}$ . New added edge heuristically inherit minimum distance sub-path around the neighborhood of the current node from two parent individuals. Meanwhile, the added node is set as the current node in  $\pi_{qi}$ . For further expanding the searching range, the above neighborhood can be set as two-step or much more steps while the calculation and the computational time will be increasing.

3) Repeating the process of inserting the minimum distance edge around the neighborhood of the current node until the neighbor nodes of the current node has already been added the sequence of  $\pi_{qi}$ . Then, the nearest node to the current node will be sought in these remaining nodes that have not been added the sequence of  $\pi_{qi}$ .

4) Repeating the above processing of the local search crossover operator until all of the nodes have been added into the sequence of  $\pi_{qi}$ . Thereby, one complete valid offspring chromosome  $\pi_{qi}$  is produced.

5) Another offspring chromosome  $\pi_{qj}$  is produced by using the same way. Finally, we produce two feasible offspring chromosomes from two parent individuals by local search crossover operator.

In every generation followed by the heuristic tour improvement, more refined indi-

viduals are exploited to moderately improve the population quality in approximating the global optimum or sub-optimum solutions as the execution progress.

### 3.4. Double-Bridge Random Mutation Operator

Double-bridge [14] random mutation operator is used to maintain the diversity of the population in HGA. It perturbs the original individual and increases the probability of generating better offspring through randomly modifying the gene values. Double-bridge random mutation operator plays an important role of avoiding the evolutionary process to trap into the local optimum and exploring an increasingly larger region of the solution space.

The HGA adequately utilizes the global exploring ability of GA and the efficiency of the local search method to obtain the satisfactory final solutions with acceptable time-consuming.

## 4. Experiments and Results

In these experiments, the computing platform is MATLAB R2012a on 2.30GHz CPU Intel Corei5-6300HQ and 4.00 GB RAM. For the comprehensive analysis of the feasibility and validity of the HGA, we carry out experiments on the two-dimensional symmetric Euclidean TSP instances from the best-known TSPLIB benchmark. Each instance is executed 20 times to obtain an average behavior for the method. The parameters of HGA are set as following: crossover probability  $P_c = 0.6$ ; mutation probability  $P_m = 0.3$ . The population size  $M$  is roughly equal to the number of cities. Those experimental results are shown in **Table 1**.

In **Table 1**, the performance indices in these experiments are described as follows:

1) Instance (Optimal): The name of instances and the corresponding length of optimal tour in parentheses. The name of each TSP instance has a suffix number that represents the number of city nodes of the instance.

**Table 1.** The experimental results of HGA.

Instance (Optimal)	Best	Ave.	Gap (%)	Gen.	Time (s)
Eil51 (426)	436	440	2.35	97	26
Berlin52 (7542)	7544	7559	0.03	134	35
Eil76 (538)	552	559	2.60	186	103
Rat99 (1211)	1242	1257	2.56	194	125
Eil101 (629)	651	661	3.50	223	178
KroA150 (26,524)	27,048	27,297	1.98	222	241
Tsp225 (3916)	3961	3991	1.15	202	552
Gil262 (2378)	2449	2481	2.99	156	812
Lin318 (42,029)	43,162	43,873	2.70	182	1214
Pr439 (107,217)	110,185	111,394	2.77	186	2264
Rat575 (6773)	7253	7285	7.09	196	4120

- 2) Best: The best tour length with the 20 independent executions.
- 3) Ave.: The total average tour length of 20 trials.
- 4) Gap (%): The relative error of the length of the best tour to that of the optimal tour, which is formulated as Equation (2):

$$Gap = \frac{L_{best} - L_{opt}}{L_{opt}} \times 100(\%) \quad (2)$$

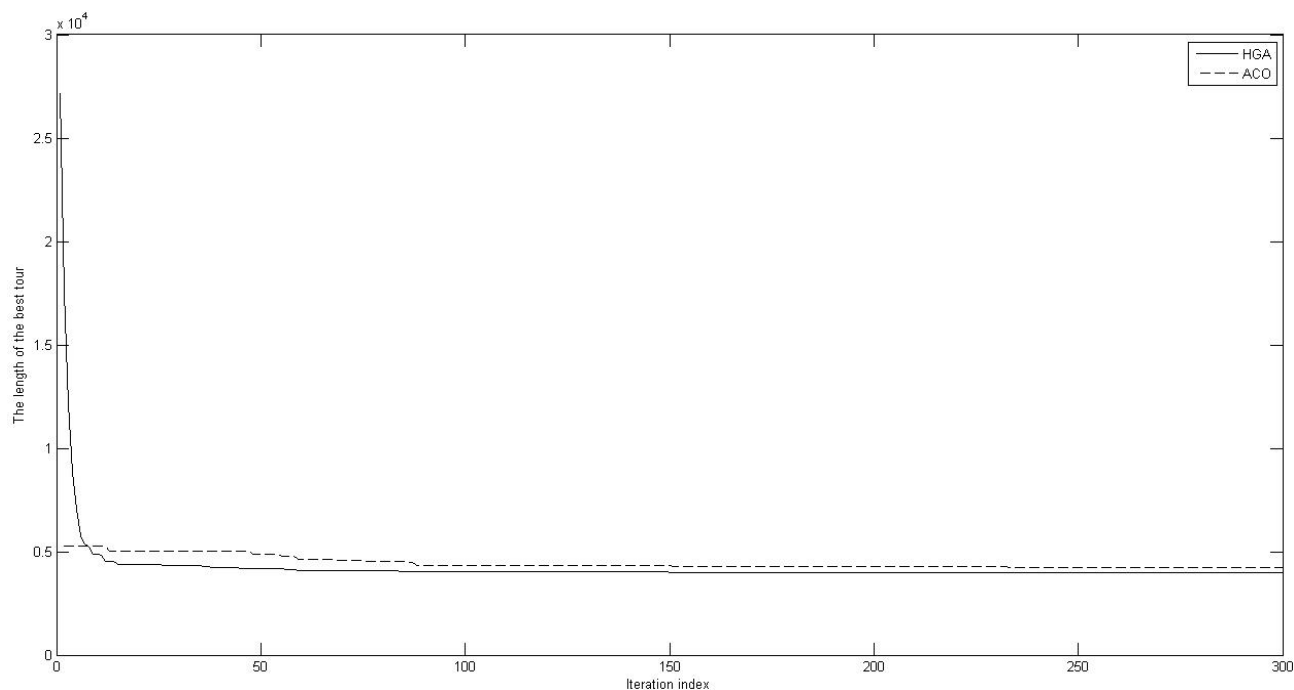
where  $L_{best}$  is the best tour length among all the 20 trials and  $L_{opt}$  is the best-known optimal tour length of each instance so far.

- 5) Gen.: The average numbers of the iteration.
- 6) Time (s): the average computational time in seconds.

By observing the figures in **Table 1**, the results show that HGA could mostly works very well on the medium-scale TSP instances with up to 575 nodes. These average solutions are also stable and able to reach close to the optimum of each instance. The relative errors between the best tour lengths and TSPLIB optimum tour lengths: For Berlin52, Tsp225 and KroA150, HGA obtained better solutions can reach to the optimum solutions with high ratios of 0.03, 1.15 and 1.98 respectively. Although the computational time depends on the different data characteristic of each instance, the time-consuming increases with the increase of the case size to solve TSP in rationale and the bigger the size of a case, the greater the computing time. As the number of cities increases such as Tsp225, Lin318, Pr439 and Rat575, the average computational time increases with exponential order such as 552, 1214, 2264 and 4120. This reason is appropriate that the computational complexity will increase as well in executing local search crossover operator. All aspects of the statistical analysis represent the performances of the HGA including the final solution quality and the computational time. In general, Experimental results demonstrate that HGA can perform effectively and efficiently on the medium-scale TSP instance.

For the comparative experiment, there is an instance Tsp225 of the specific evolutionary processing obtained by HGA and the traditional ACO as shown in **Figure 1**. The diagram plots the lengths of the best tour obtained by those methods as a function of the iteration index.

For the fairness of the comparative results, the final best solution of each algorithm is allowed to evolve sufficiently. As it can be seen, the macroscopic behavior of those algorithms is presented by subsequently convergence rate and solution quality. For example, HGA converges at less than 150 iterations while the traditional ACO execute about 235 iterations to achieve the same or worse results. Meanwhile, HGA can continue to converge to a better solution due to the local search ability of the competitive updating strategy. In addition, HGA performs 202 iterations on average and returns the final best solution with the relative error of only 1.15% from the optimal, while the traditional ACO executes about 247 iterations on average but returns the final best solutions with the relative error of 8.27% that is more than 7 times higher. The average tour length of HGA is 3991 while that of the traditional ACO is 4240 in the Tsp225 instance.



**Figure 1.** The comparative experiment of HGA and ACO.

Therefore, HGA performs more effectively and efficiently than the traditional ACO in terms of the convergence rate and the solution quality.

For further to validate the effectiveness of HGA for TSP, we perform the comparative experiments to assess the contributions of HGA and other different algorithms. **Table 2** shows the comparative results of HGA and those of the GA introduced in the article [9].

The results in **Table 2** indicate that the average best tour lengths and the final tour lengths found by HGA are better than those obtained the GA [9] on the majority of instances. Meanwhile, the relative error of the best solution found by HGA is very higher than that of the GA [9] on most of instances. HGA has better solution quality than the GA [9] but suffers from a slightly slow speed. With the increase in the case size of TSP, we test HGA and the GA introduced in the article [10] on these TSP benchmarks with up to 442 cities. **Table 3** also shows the comparative experimental results of HGA and the GA [10].

The results in **Table 3** show that these improvements in the solution quality are statistically significant. The average solutions, the final solutions and the relative errors found by HGA are better than those obtained the GA [10] on these instances. Therefore, HGA outperforms those Genetic Algorithms in terms of the solution quality. HGA is a heuristic hybrid evolutionary algorithm with efficiency and effectiveness for solving TSP.

## 5. Conclusion

In this paper, a novel heuristic HGA integrating GA and the local search method in a



**Table 2.** The comparative results of GA [9] and HGA.

Instance(Optimal)	GA [9]				HGA			
	Best	Ave.	Gap (%)	Time (s)	Best	Ave.	Gap (%)	Time (s)
Eil51(426)	438	449	2.82	17	436	440	2.35	26
Berlin52(7542)	7542	8024	0.00	19	7544	7559	0.03	35
St70(675)	700	713	3.70	39	683	686	1.19	81
Eil76(538)	565	577	5.02	50	552	559	2.60	103
KroB100(22141)	22,722	23,195	2.62	115	22,426	22,596	1.29	172
Eil101(629)	670	690	2.62	115	651	661	3.50	178
Pr107(44303)	44,678	45,584	0.85	127	44,661	45,150	0.81	140
Pr124(59030)	59,199	61,041	0.29	218	59,452	59,784	0.71	206

**Table 3.** The comparative results of GA [10] and HGA.

Instance(Optimal)	GA [10]			HGA		
	Best	Ave.	Gap (%)	Best	Ave.	Gap (%)
KroA100(21282)	23,909	24,270	12.34	21,733	21,811	2.12
KroA150 (26524)	28,651	29,418	8.02	27,048	27,297	1.98
KroA200 (29368)	32,646	33,978	11.16	30,014	30,166	2.20
Pr299(48191)	54,104	55,938	12.27	50,419	50,735	4.62
Pcb442(50778)	53,849	55,798	6.05	53,849	54,371	6.05

good symbiotic behavior is proposed and illustrated for the two-dimensional symmetric Euclidean TSP. The proposed HGA provides much better reasonable structure design to compensate for the balance between global and local search contributes to efficiently improve the evolutionary process in terms of the convergence rate and the solution quality. Experimental results demonstrate that HGA is a robust hybrid evolutionary algorithm to search the global optimum or sub-optimum solutions within reasonable computational time. Meanwhile, HGA works very well on the medium-scale TSP instances and outperforms the other GAs in the 4 section. As future research work, there is a lot of scope for development in HGA with large-scale TSP and their applications. We also plan to investigate extensions of the HGA for other complex optimization problems.

## Acknowledgements

This work is jointly supported by the National Natural Science Foundation of China with grants No. 61473298 and No. 61473299.

## References

- [1] Lawler, E.L.G., Lenstra, J.K., Rinnooy Kan, A.H.G. and Shmoys, D.B. (1985) The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. *ser. Estimation, Simu-*

- lation, and Control—Wiley-Interscience Series in Discrete Mathematics and Optimization.* Wiley Interscience, Chichester, West Sussex, UK.
- [2] Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* University Michigan Press, Ann Arbor, MI, USA.
- [3] Grefenstette, J., Gopal, R., Rosinaita, B. and van Gucht, D. (1985) Genetic Algorithms for the Traveling Salesman Problem. *Proc. Int. Conf. Genetic Algorithms and Their Applications*, July 1985, 160-168.
- [4] Kirkpatrick, S., Gelatt Jr., C.D. and Vecchi, M.P. (1983) Optimization by Simulated Annealing. *Science*, **220**, 498-516. <http://dx.doi.org/10.1126/science.220.4598.671>
- [5] Dorigo, M., Maniezzo, V. and Coloni, A. (1996) The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man and Cybernetics Part B*, **26**, 29-41. <http://dx.doi.org/10.1109/3477.484436>
- [6] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*, **4**, 1942-1948. <http://dx.doi.org/10.1109/icnn.1995.488968>
- [7] Glover, F. (1989) Tabu Search, Part I. *ORSA Journal on Computing*, **1**, 190-206. <http://dx.doi.org/10.1287/ijoc.1.3.190>
- [8] Potvin, J.Y. (1993) The Traveling Salesman Problem: A Neural Network Perspective. *ORSA J. Comput.*, **5**, 328-348. <http://dx.doi.org/10.1287/ijoc.5.4.328>
- [9] Osaba, E., Carballedo, R., Diaz, F. and Perallos, A. (2013) Analysis of the Suitability of Using Blind Crossover Operators in Genetic Algorithms for Solving Routing Problems. *IEEE International Symposium on Applied Computational Intelligence and Informatics*, 17-22.
- [10] Chang, P.C., Huang, W.H., Ting, C.J. and Chang, W.J. (2009) A Varietal Genetic Algorithm by External Self-evolving Multiple-archives for Combinatorial Optimization Problems. *Proceedings of the 11th IEEE International Conference on High Performance Computing and Communications*, 609-614. <http://dx.doi.org/10.1109/hpcc.2009.67>
- [11] Wu, Y., Weise, T. and Chiong, R. (2015) Local Search for the Traveling Salesman Problem: A Comparative Study. *Proceedings of the 14th IEEE Int'l Conf. on Cognitive Informatics & Cognitive Computing*, 213-220. <http://dx.doi.org/10.1109/icci-cc.2015.7259388>
- [12] Pham, D.T. and Huynh, T.T.B. (2015) An Effective Combination of Genetic Algorithms and the Variable Neighborhood Search for Solving Travelling Salesman Problem. *IEEE Conference on Technologies and Applications of Artificial Intelligence*, 142-149.
- [13] Reinelt, G. (1991) TSPLIB—A Traveling Salesman Problem Library. *ORSA Journal on Computing*, **3**, 376-384. <http://dx.doi.org/10.1287/ijoc.3.4.376>
- [14] Helsgaun, K. (2009) General k-opt Submoves for the Lin-Kernighan TSP Heuristic. *Mathematical Programming Computation*, **1**, 119-163. <http://dx.doi.org/10.1007/s12532-009-0004-6>



**Submit or recommend next manuscript to SCIRP and we will provide best service for you:**

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact [jcc@scirp.org](mailto:jcc@scirp.org)